



06-16-04

IFW

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:  
**Robert Denk**

Serial No.: 10/810,531

Filing Date: March 26, 2004

Title: **Method and Apparatus for  
Determination of Initialization States  
in Pseudo-Noise Sequences**

§  
§  
§  
§  
§  
§  
§  
§  
§

Group Art Unit: 2631

Examiner:

Attny. Docket No. 068758.0181

Client Ref.: 10290US/lg

Mail Stop Missing  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

## CERTIFICATE OF MAILING VIA EXPRESS MAIL

PURSUANT TO 37 C.F.R. § 1.10, I HEREBY CERTIFY THAT I HAVE INFORMATION AND A REASONABLE BASIS FOR BELIEF THAT THIS CORRESPONDENCE WILL BE DEPOSITED WITH THE U.S. POSTAL SERVICE AS EXPRESS MAIL POST OFFICE TO ADDRESSEE, ON THE DATE BELOW, AND IS ADDRESSED TO:

MAIL STOP  
COMMISSIONER FOR PATENTS  
P.O. Box 1450  
ALEXANDRIA, VA 22313-1450

EXPRESS MAIL LABEL: EV448723621US  
DATE OF MAILING: JUNE 15, 2004

## SUBMISSION OF PRIORITY DOCUMENT

Dear Sir:

We enclose herewith a certified copy of German patent application 101 47 306.0 which is the priority document for the above-referenced patent application.

Respectfully submitted,

BAKER BOTTS L.L.P. (023640)

Date: June 15, 2004

By:   
Andreas H. Grubert  
(Limited recognition 37 C.F.R. §10.9)  
One Shell Plaza  
910 Louisiana Street  
Houston, Texas 77002-4995  
Telephone: 713.229.1964  
Facsimile: 713.229.7764  
AGENT FOR APPLICANTS

# BUNDESREPUBLIK DEUTSCHLAND

---



## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 101 47 306.0

**Anmeldetag:** 26. September 2001

**Anmelder/Inhaber:** Infineon Technologies AG, 81669 München/DE

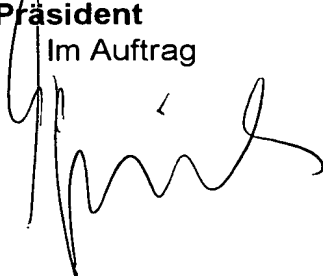
**Bezeichnung:** Verfahren und Vorrichtung zur Bestimmung von Initialisierungszuständen bei Pseudo-Noise-Folgen

**IPC:** H 04 J, H 04 B

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 26. April 2004  
**Deutsches Patent- und Markenamt**  
**Der Präsident**

Im Auftrag

 **Agurks**

## Beschreibung

Verfahren und Vorrichtung zur Bestimmung von Initialisierungszuständen bei Pseudo-Noise-Folgen

5

Die Erfindung betrifft ein Verfahren sowie eine Vorrichtung zur Bestimmung eines  $n$  Bit umfassenden,  $N$ -fach iterierten Endzustands einer Schieberegisteranordnung aus einem gegebenen,  $n$  Bit umfassenden Anfangszustand der Schieberegisteranordnung. Des weiteren betrifft die Erfindung die Erzeugung von um  $N$  Bit verschobenen Pseudo-Noise-Folgen, welche insbesondere als Spreizfolgen in CDMA-basierten Mobilfunksystemen (CDMA: Code Division Multiple Access) verwendet werden.

15

In einem Mobilfunksystem werden die von der Basisstation bzw. von der Mobilstation generierten Signale vor ihrer Aussendung mehrere Male modifiziert. Unter anderem, um verschiedene Zellen in einem Mobilfunknetz unterscheiden zu können, werden bei CDMA-Systemen Spreizfolgen eingesetzt, wobei jedem Benutzer und jedem logischen Kanal eine andere Folge der Werte  $-1$  und  $1$  zugeordnet wird. Dadurch kann das dem einzelnen Benutzer zugeordnete Signal empfangen, von den anderen Signalen getrennt und rekonstruiert werden.

25

Dies bezeichnet man als Code Division Multiple Access (CDMA). Im Gegensatz dazu werden bei TDMA-Systemen (Time Division Multiple Access) die Signale zeitlich voneinander getrennt. Wichtige CDMA-Übertragungssysteme sind das in den USA verwendete System IS-95 und das System UMTS, welches im 3<sup>rd</sup> Generation Partnership Project (3GPP) spezifiziert wird. Die genaue Beschreibung der verwendeten Codierung für UMTS ist in "3GPP: Spreading and modulation (FDD)", 3<sup>rd</sup> Generation Partnership Project TS 25.213, Release 1999 zu finden.

35

Alle verwendeten Spreizcodes lassen sich auf Folgen der Binärwerte  $0$  und  $1$  zurückführen. Bei diesen Folgen kann es

sich beispielsweise um sogenannte Pseudo-Noise-Folgen handeln, welche durch definierte Autokorrelations- und Kreuzkorrelationseigenschaften gekennzeichnet sind. Während in der theoretischen Darstellung eine Pseudo-Noise-Folge als Folge der Binärwerte 0 und 1 dargestellt wird, handelt es sich bei der tatsächlich verwendeten Spreizfolge um eine Folge der Werte +1 und -1. Aus dem Binärwert 0 wird in der tatsächlichen Spreizfolge jeweils der Wert +1.

Pseudo-Noise-Folgen sind durch eine Iterationsvorschrift definiert, wobei die Iteration im Körper  $GF(2)$ , also im Zahlkörper mit den beiden Elementen 0 und 1, ausgeführt wird. Theoretische Grundlage der Pseudo-Noise-Folgen und der definierenden Iterationsvorschrift ist die Theorie irreduzibler primitiver Polynome über dem Körper  $GF(2)$ . Eine Darstellung dieser Theorie und ihrer Anwendung im Mobilfunkbereich findet sich zum Beispiel in "CDMA Systems Engineering Handbook" von J.S. Lee, L.E. Miller, Artech House, Boston/London, 1998, und dort insbesondere in Kapitel 6.

Jede individuelle Pseudo-Noise-Folge ist eindeutig festgelegt durch den Anfangszustand, also durch die ersten Werte der Folge, sowie durch das für die Iteration verwendete Polynom. Dabei ist das Polynom und damit die Iterationsvorschrift in Mobilfunkanwendungen entweder für das gesamte Netzwerk festgelegt, oder es kommen insgesamt nur einige wenige verschiedene Polynome zur Anwendung, wie dies zum Beispiel bei UMTS-Systemen für die Definition der sogenannten Scrambling-Codes der Fall ist. Der Anfangszustand ist für jede individuelle Pseudo-Noise-Folge verschieden und wird häufig durch die Code-Nummer festgelegt.

In einer Basisstation bzw. in einer Mobilstation muss daher zu einer gegebenen Code-Nummer und zu einer ebenfalls vorgegebenen Iterationsvorschrift die zugehörige Pseudo-Noise-Folge generiert werden. Beim Sendebetrieb muss die erzeugte Folge zur Codierung des Signals verwendet werden. Im Emp-

fangsmodus hingegen erlaubt die Verwendung der Pseudo-Noise-Folge, das gewünschte Signal zu erkennen und von den Signalen für andere Benutzer zu unterscheiden. Falls die Anfangswerte der gesuchten Folge bekannt sind, können die weiteren Folgenwerte durch einfache Register-Operationen ohne Schwierigkeiten erzeugt werden. Dabei muss auf die zeitliche Übereinstimmung zwischen der zu sendenden bzw. der empfangenen Information einerseits und der konstruierten Folge andererseits geachtet werden.

10

In verschiedenen Mobilfunkanwendungen ist jedoch der Anfang der Folge und damit der Anfangszustand der Register nicht bekannt. Dies ist zum Beispiel dann der Fall, wenn die Codierung zu einem anderen Zeitpunkt gestartet werden soll als die Signalübertragung selbst. Dieser Fall tritt bei UMTS im sogenannten Compressed Mode auf; nähere Informationen zu diesem Modus finden sich in "3GPP: Physical channels and mapping of transport channels onto physical channels (FDD)", 3<sup>rd</sup> Generation Partnership Project TS 25.211, Release 1999.

20

Der Anfang der Folge und damit der Anfangszustand der Register ist auch dann nicht bekannt, wenn die Code-Nummer nicht direkt die anfängliche Registerbelegung festlegt, sondern statt dessen eine Verschiebung der verwendeten Pseudo-Noise-Folge um eine gewisse Anzahl von Bits definiert. So wird zum Beispiel in UMTS beim Empfang eines Signals im Mobilteil entsprechend dem 3GPP-Standard der Code Nummer N als eine um N Bits verschobene Pseudo-Noise-Folge definiert. Nähere Informationen zu dem Zusammenhang zwischen der Code-Nummer und der zugehörigen Pseudo-Noise-Folge finden sich in "3GPP: Spreading and modulation (FDD)", 3<sup>rd</sup> Generation Partnership Project TS 25.213, Release 1999, und zwar insbesondere in Abschnitt 5.2.

35

Um den Anfangszustand der Register für den Fall zu berechnen, dass die Folge einer zusätzlichen Verschiebung bzw.

5 einem zusätzlichen Offset von N Bits unterworfen wurde, kann die Folge zum ursprünglichen Anfangszeitpunkt gestartet und anschließend N-fach iteriert werden. Auf diese Weise kann die gewünschte, um N Bits verschobene Folge erhalten werden.

In bisherigen Systemen des Stands der Technik wurde diese Lösung praktiziert. Vor der Ausgabe der gewünschten Pseudo-Noise-Folge wurde der Registerinhalt der Schieberegisterstruktur N-fach iteriert. Erst nach der Durchführung dieser Vorab-Iterationen wurde mit der Ausgabe der eigentlichen, um N Bits verschobenen Pseudo-Noise-Folge begonnen. Ein Nachteil dieser Vorgehensweise ist, dass die Anzahl der benötigten Operationen proportional zur Größe der gewünschten Verschiebung N ist. Die Anzahl der benötigten Operationen variiert daher in Abhängigkeit von den aktuellen Daten, und dies erschwert die Kontrolle der zeitlichen Gesamtabfolge. Ein weiterer Nachteil ist, dass bei großen Werten der gewünschten Verschiebung N der rechnerische und zeitliche Aufwand sehr groß wird. Bei UMTS-Systemen treten im Empfangsbetrieb der Mobilstation Offsets im Bereich von  $N = 0$  bis  $N = 262142$  auf. Da mit der Erzeugung der gewünschten Pseudo-Noise-Folge gewartet werden muss, bis der gewünschte Offset erreicht ist, bedeutet dies eine inakzeptable Verzögerung des Sende- bzw. Empfangsvorgangs.

Es ist daher Aufgabe der Erfindung, zu einem gegebenen Anfangszustand den N-fach iterierten Endzustand bzw. die um N Bit verschobene Pseudo-Noise-Folge direkt zu berechnen.

30 Diese Aufgabe der Erfindung wird durch ein Verfahren zur Bestimmung eines n Bit umfassenden, N-fach iterierten Endzustands nach Anspruch 1, durch eine Vorrichtung zur Bestimmung eines n Bit umfassenden, N-fach iterierten Endzustands nach Anspruch 14 sowie durch die Verwendung dieser Vorrichtung zur Erzeugung einer Spreizfolge gemäß Anspruch 35 20 gelöst.

Bei dem erfindungsgemäßen Verfahren zur Bestimmung eines  $n$  Bit umfassenden,  $N$ -fach iterierten Endzustands einer Schieberegisteranordnung aus einem gegebenen,  $n$  Bit umfassenden Anfangszustand der Schieberegisteranordnung ist die Iterationsvorschrift durch das charakteristische Polynom

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

mit  $c_1, c_2, \dots, c_{n-1} \in \{0; 1\}$  gegeben. In einem ersten Schritt wird das Polynom

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

durch Spiegelung der Koeffizienten des Polynoms

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

bestimmt. Anschließend wird für  $j = 1, \dots, n$  derjenige Vertreter der Restklasse

$$\left[ x^{N+j-1} \right] \bmod f^*$$

bestimmt, dessen Grad kleiner als  $n$  ist. Daraufhin wird die Bitsequenz des Anfangszustands mit einer Matrix multipliziert, deren  $j$ -te Zeile bzw.  $j$ -te Spalte für  $j = 1, \dots, n$  durch die Koeffizienten des Vertreters der Restklasse

$$\left[ x^{N+j-1} \right] \bmod f^*$$

gegeben ist.

Mit dem erfindungsgemäßen Verfahren ist es erstmals möglich, den Zustand einer durch ein charakteristisches Polynom definierten Schieberegisteranordnung, welcher sich nach Durchführung von  $N$  Iterationen ergibt, explizit zu berechnen. Im Stand der Technik mussten zur Bestimmung dieses

Endzustands N Iterationen der Schieberegisteranordnung vorab ausgeführt werden. Da zur Erzeugung der verschiedenen bei der Mobilfunkübertragung benötigten Codes teilweise bis zu  $N = 262142$  Vorab-Iterationen ausgeführt werden müssten, ist durch die Möglichkeit, den N-fach iterierten Endzustand explizit zu berechnen, eine immense Zeitersparnis gegeben. Die Erfindung bietet die Möglichkeit, eine bestimmte, um den Offset N verschobene Codefolge quasi verzögerungsfrei zur Verfügung zu stellen.

10

Die Bestimmung von demjenigen Vertreter der Restklasse

$$\left[ x^{N+j-1} \right] \bmod f^*$$

15

dessen Grad kleiner als  $n$  ist, kann mit Hilfe von schnellen Algorithmen zur Restklassenberechnung, beispielsweise mit Hilfe von Square-and-Multiply Algorithmen, in sehr kurzer Zeit durchgeführt werden. Für den rechnerischen und zeitlichen Aufwand zur Bestimmung eines Vertreters der Restklasse

20

$$\left[ x^{N+j-1} \right] \bmod f^*$$

25

ergibt sich dabei eine logarithmische Abhängigkeit von  $N$ , also eine logarithmische Abhängigkeit von der gewünschten Offset-Verschiebung der Codefolge. Bei der Durchführung von  $N$  Vorab-Iterationen, wie sie im Stand der Technik notwendig war, nahm der rechnerische und zeitliche Aufwand zur Durchführung der Vorab-Iterationen linear mit  $N$  zu. Insbesondere für große Werte von  $N$  ergibt sich bei der erfindungsgemäßen Lösung wegen der logarithmischen Abhängigkeit von  $N$  eine enorme Verkürzung der benötigten Rechenzeit.

30

Durch Schieberegisteranordnungen erzeugte Pseudo-Noise-Folgen werden insbesondere zur senderseitigen Codierung sowie zur empfängerseitigen Decodierung von Datenpaketen bei der Mobilfunkübertragung benötigt. Bei bisherigen Lösungen wurde durch die Durchführung von  $N$  Vorab-Iterationen eine

35



inakzeptable Verzögerung des Sende- bzw. des Empfangsvorgangs verursacht. Derartig Verzögerungen können bei der erfindungsgemäßen Lösung vermieden werden, weil hier der N-fach iterierte Endzustand mittels einer Matrixmultiplikation und nicht wie bisher iterativ bestimmt wird.

Codefolgen, welche um N Bits verschoben sind, werden entsprechend dem 3GPP-Standard mit der Code-Nummer N bezeichnet. Die Erfindung bietet daher die Möglichkeit, sämtliche im 3GPP-Standard für die Mobilfunkübertragung definierten Codes ohne Wartezeit zu erzeugen. Bei der Funkübertragung von codierten Signalen gibt es darüber hinaus den Fall, dass die Codierung zu einem anderen Zeitpunkt gestartet werden soll als die Signalübertragung selbst. Beim Mobilfunkstandard UMTS ist dies beispielsweise beim sogenannten Compressed Mode der Fall. Mit Hilfe des erfindungsgemäßen Verfahrens, welches instantan einen N-fach iterierten Zustand der Schieberegisteranordnung erzeugen kann, kann somit der richtige Anfangszustand der Schieberegisteranordnung für die Signalcodierung im Compressed Mode generiert werden.

Die Erfindung eignet sich für alle Anwendungen, bei denen Codefolgen mit Hilfe einer getakteten, rückgekoppelten Schieberegisteranordnung erzeugt werden. Dabei wird die in der Schieberegisteranordnung vorgesehene Rückkopplung durch das charakteristische Polynom

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

festgelegt. Der n Bit umfassende Schieberegisterinhalt wird durch ein Taktsignal durch die Schieberegisteranordnung geschoben, wobei Bits, die aus dem Schieberegister herausgeschoben werden, zum Eingang der Schieberegisteranordnung rückgekoppelt werden. Derartige Schieberegisteranordnungen werden zum Zweck der Codierung und Decodierung eingesetzt. Mit dem erfindungsgemäßen Verfahren lässt sich zu einem ge-

gebenen Anfangszustand der Schieberegisteranordnung der Endzustand berechnen, der nach N Schiebeoperationen beziehungsweise nach N an das Schieberegister angelegten Taktimpulsen erreicht ist.

5

Die erfindungsgemäße rechnerische Bestimmung des Endzustands erfordert die Berechnung einer Matrix sowie die Multiplikation des Anfangszustands mit dieser Matrix. Die rechnerische Bestimmung der Matrixelemente sowie die Durchführung der Matrixmultiplikation kann dabei von einem Prozessor, insbesondere von einem digitalen Signalprozessor, durchgeführt werden. Der berechnete Endzustand kann dann zur Initialisierung der im Hardware realisierten Schieberegisteranordnung verwendet werden. Die Erfindung ermöglicht es, die verschiedenen zur Code-Erzeugung benötigten Initialisierungszustände mit geringem rechnerischen Aufwand zuverlässig zu bestimmen.

Dabei ist es von Vorteil, wenn die Vertreter der Restklassen

$$[x^N] \bmod f^*, [x^{N+1}] \bmod f^*, \dots [x^{N+n-1}] \bmod f^*$$

jeweils explizit mittels eines geeigneten Algorithmus, insbesondere mittels eines Square-and-Multiply Algorithmus berechnet werden. Für die Berechnung der Restklassen

$$[x^m] \bmod f^*,$$

von Monomen, wobei m eine natürliche Zahl ist, existieren eine Reihe verschiedener Algorithmen, welche jeweils die Koeffizienten von demjenigen Vertreter der Restklasse liefern, dessen Grad kleiner ist als n. Der rechnerische und zeitliche Aufwand bei der Ausführung dieser Algorithmen hängt dabei logarithmisch von m ab. Die zur Durchführung des erfindungsgemäßen Verfahrens benötigten Matrixelemente

können daher auch für große Werte von  $N$  schnell erzeugt werden.

Dabei ist es insbesondere von Vorteil, wenn ein Square-and-Multiply Algorithmus eingesetzt wird. Ausgehend von dem Vertreter der Restklasse

$$[x] \bmod f^*$$

10 kann mit einem Square-and-Multiply Verfahren sehr schnell der Vertreter von

$$[x^m] \bmod f^*$$

15 berechnet werden, wobei  $m$  eine natürliche Zahl ist. Ein derartiger Square-and-Multiply Algorithmus wird in der Beschreibung dieser Patentanmeldung explizit aufgeführt. Der Algorithmus besteht nur aus wenigen Zeilen, lässt sich einfach implementieren und liefert verlässliche Ergebnisse für  
20 die Koeffizienten des Vertreters der Restklasse

$$[x^m] \bmod f^*.$$

Gemäß einer vorteilhaften Ausführungsform der Erfindung  
25 wird lediglich der Vertreter der Restklasse

$$[x^N] \bmod f^*$$

explizit mittels eines geeigneten Algorithmus, insbesondere  
30 mittels eines Square-and-Multiply Algorithmus, berechnet. Die Vertreter der Restklassen

$$[x^{N+j-1}] \bmod f^*$$

35 mit  $j = 2, \dots, n$  werden dagegen durch  $(n-1)$  rechnerisch durchgeführte Iterationen aus den Koeffizienten des Vertreters der Restklasse

$$\left[ x^N \right] \bmod f^*$$

erhalten. Anstatt für alle  $N$  Zeilen der zu bestimmenden  
 5 Matrix die Vertreter

$$\left[ x^{N+j-1} \right] \bmod f^*$$

mit Hilfe eines Square-and-Multiply Algorithmus zu bestim-  
 10 men, wird bei dieser Ausführungsform der Erfindung der  
 Square-and-Multiply Algorithmus nur noch für die erste Zei-  
 le der Matrix durchgeführt. Die Matrixelemente der restli-  
 chen  $(n-1)$  Zeilen der Matrix werden dann mit Hilfe von  
 $(n-1)$  rechnerisch durchgeführten Iterationen dieser Koeffi-  
 15 zienten erzeugt. Aus den Matrixelementen der  $j$ -ten Zeile  
 können jeweils die Matrixelemente der  $(j+1)$ -ten Zeile be-  
 stimmt werden. Der Vorteil dieser Vorgehensweise gegenüber  
 dem  $n$ -maligen Aufruf des Algorithmus ist eine weitere rech-  
 nerische Vereinfachung bei der Bestimmung der Matrixelemen-  
 20 te. Die Zahl der zur Bestimmung der Matrixelemente erfor-  
 derlichen Rechenschritte wird weiter verringert, und damit  
 ist die Berechnung des  $N$ -fach iterierten Endzustands in  
 noch kürzerer Zeit möglich.

25 Dabei ist es von Vorteil, wenn die Vertreter der Restklas-  
 sen

$$\left[ x^{N+j-1} \right] \bmod f^*$$

30 mit  $j = 2, \dots, n$  durch  $(n-1)$  rechnerisch durchgeführte Ite-  
 rationen einer Schieberegisteranordnung vom MSRG-Typ  
 (Modular Shift Register Generator) aus den Koeffizienten  
 des Vertreters der Restklasse

35  $\left[ x^N \right] \bmod f^*$

erhalten werden, wobei die Iterationsvorschrift für die Schieberegisteranordnung durch das charakteristische Polynom

$$5 \quad f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

gegeben ist.

Um aus den Koeffizienten des Vertreters der Restklasse

$$10 \quad [x^{N+j-1}] \bmod f^*$$

die Koeffizienten des Vertreters der Restklasse

$$15 \quad [x^{N+j}] \bmod f^*$$

zu erhalten, um also aus der  $j$ -ten Zeile die  $(j+1)$ -te Zeile herzuleiten, wird eine rechnerische Iteration dieser Koeffizienten durchgeführt, die dem Durchschieben dieser Koeffizienten durch ein Schieberegister vom MSRG-Typ entspricht. Die Struktur eines Schieberegisters vom Typ MSRG wird durch das charakteristische Polynom

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

25 festgelegt. Allerdings wird die iterative Bestimmung der Matrixelemente in der Regel nicht von einer in Hardware realisierten Schieberegisteranordnung durchgeführt, sondern rein rechnerisch mit Hilfe von Software oder mit Hilfe eines Prozessors, etwa eines digitalen Signalprozessors.

Die explizite Berechnung der ersten Matrixzeile, also der Koeffizienten des Vertreters der Restklasse

$$35 \quad [x^N] \bmod f^*,$$

und die iterative Herleitung der restlichen Koeffizienten stellt die einfachste und schnellste Möglichkeit zur Berechnung sämtlicher Matrixelemente dar.

- 5 Es ist von Vorteil, wenn der  $n$  Bit umfassende,  $N$ -fach iterierte Endzustand als Initialisierungszustand für die Erzeugung einer um  $N$  Bit verschobenen Pseudo-Noise-Folge verwendet wird. Eine Folge von Binärwerten, welche von einer rückgekoppelten, durch ein irreduzibles Polynom beschriebenen Schieberegisteranordnung erzeugt wird, wird als Pseudo-Noise-Folge bezeichnet. Eine Pseudo-Noise-Folge wird zum  
10 einen durch den Initialzustand der Schieberegisteranordnung und zum anderen durch das charakteristische Polynom der Schieberegisteranordnung festgelegt. Wenn der mittels des  
15 erfindungsgemäßen Verfahrens berechnete,  $N$ -fach iterierte Endzustand als Initialisierungszustand für die Erzeugung einer Pseudo-Noise-Folge verwendet wird, dann bedeutet dies, dass die Pseudo-Noise-Folge sofort an der gewünschten, um  $N$  Bit verschobenen Stelle gestartet werden kann.  
20 Ausgehend vom Initialisierungszustand werden dann die weiteren Folgenwerte geliefert.

- Es ist von Vorteil, wenn der  $n$  Bit umfassende,  $N$ -fach iterierte Endzustand als Initialisierungszustand in eine  $n$   
25 Schieberegisterzellen umfassende Schieberegisteranordnung eingeschrieben wird. Mit Hilfe des erfindungsgemäßen Verfahrens wird der  $N$ -fach iterierte Endzustand rechnerisch bestimmt und anschließend in die in Hardware realisierte Schieberegisteranordnung eingeschrieben. Da der rechnerisch  
30 bestimmte,  $N$ -fach iterierte Endzustand genau dem Zustand der Schieberegisteranordnung nach der Durchführung von  $N$  Iterationen entspricht, kann ausgehend von dem berechneten Initialisierungszustand die gewünschte, um  $N$  Bit verschobene Pseudo-Noise-Folge erzeugt werden. Nach dem Einschreiben  
35 des berechneten Initialisierungszustands in die Schieberegisteranordnung ist nicht mehr erkennbar, ob dieser Zustand

mit Hilfe von N Vorab-Iterationen der Schieberegisteranordnung oder durch Berechnung erreicht wurde.

Er ist von Vorteil, wenn es sich bei der Schieberegisteranordnung um eine n Schieberegisterzellen umfassende Schieberegisteranordnung vom Typ SSRG handelt, deren Struktur durch das charakteristische Polynom

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

10

gegeben ist. Bei der Realisierung einer Schieberegisteranordnung in Hardware ist eine Schieberegisteranordnung vom Typ SSRG (Simple Shift Register Generator) gegenüber dem Typ MSRG (Modular Shift Register Generator) von Vorteil, weil beim Typ SSRG der Inhalt einer Schieberegisterzelle direkt in die nächstfolgende Schieberegisterzelle geschoben wird. Beim Typ MSRG dagegen sind zwischen die einzelnen Schieberegisterzellen XOR-Gatter geschaltet, welche den Inhalt einer Registerzelle beim Weiterschieben zur nächstfolgenden Registerzellen modifizieren. Bei Schieberegistern des Typs SSRG werden die Inhalte der Registerzellen nicht modifiziert, und deshalb lassen sich derartige Schieberegisteranordnungen auf einfache Weise als Array von Registerzellen implementieren.

25

Die von der Schieberegisteranordnung erzeugte Pseudo-Noise-Folge kann an der letzten Registerzelle der Schieberegisteranordnung abgegriffen werden. Mit jedem Taktimpuls, mit dem die Inhalte der Schieberegisteranordnung weitergeschoben werden, wird ein neuer Binärwert in die letzte Registerzelle der Schieberegisteranordnung geschrieben. Durch Auslesen der letzten Registerzelle der Schieberegisteranordnung erhält man daher entsprechend der Taktfrequenz, mit der die Schieberegisteranordnung getaktet wird, nacheinander die verschiedenen Folgenwerte der Pseudo-Noise-Folge.

35

Es ist von Vorteil, wenn das Verfahren in CDMA-Übertragungssystemen, insbesondere in CDMA-Übertragungssystemen mit den Übertragungsstandards UMTS oder IS-95, zur Erzeugung einer Spreizfolge mit einem Offset von N Bits verwendet wird. Pseudo-Noise-Folgen, welche sich mit Hilfe rückgekoppelter Schieberegisteranordnungen erzeugen lassen, eignen sich wegen ihrer Korrelationseigenschaften hervorragend als Spreizfolgen für CDMA-basierte Systeme, insbesondere für Mobilfunksysteme. Spreizfolgen sind endliche Folgen der Werte -1 und 1. Beim Senden einer Datenfolge wird jeder Wert der Datenfolge mit der Spreizfolge multipliziert. Empfängerseitig können dann die verschiedenen Signale anhand ihrer aufgeprägten Spreizcodierung unterschieden und selektiv decodiert werden.

Um die spreizcodierten Signale empfängerseitig eindeutig decodieren zu können, müssen die verwendeten Spreizfolgen definierte Autokorrelationseigenschaften aufweisen. Außerdem muss eine gute Unterscheidbarkeit von Signalen, welche mit verschiedenen Spreizfolgen codiert worden sind, möglich sein. Hierzu müssen die verschiedenen für die Signalübertragung verwendeten Spreizcodes definierte Kreuzkorrelationseigenschaften aufweisen. Sowohl hinsichtlich der Autokorrelationseigenschaften als auch hinsichtlich der Kreuzkorrelationseigenschaften sind Pseudo-Noise-Folgen geeignet, als Spreizfolgen eingesetzt zu werden. Spreizfolgen können daher in CDMA-Übertragungssystemen mit Hilfe von rückgekoppelten Schieberegisteranordnungen erzeugt werden.

Mit Hilfe des erfindungsgemäßen Verfahrens lassen sich Initialisierungszustände erzeugen, die es ermöglichen, dass bei der Ausgabe der Spreizfolge nicht mit dem ersten Folgenwert, sondern mit dem N-ten Folgenwert begonnen wird. Die Erfindung erlaubt also die Erzeugung von um N Bits verschobenen Spreizfolgen, also von Spreizfolgen, die einen Offset von N Bits aufweisen.



Gemäß einer vorteilhaften Ausführungsform der Erfindung wird das Verfahren zur Erzeugung der verschiedenen im UMTS-Standard definierten Scrambling-Codes eingesetzt.

Scrambling-Codes sind Spreizfolgen, die unter anderem zur  
5 Unterscheidung der Signale dienen, welche von verschiedenen Basisstationen an eine Mobilstation gesendet werden. Die erfindungsgemäße Lösung eignet sich zur Erzeugung von um N Bits verschobenen Scrambling-Codes, also von Scrambling-Codes, welche einen Offset von N Bits aufweisen. Die erfindungsgemäße Lösung ermöglicht es, ad hoc eine Vielzahl unterschiedlicher Scrambling-Codes zu generieren.  
10

Gemäß einer vorteilhaften Ausführungsform der Erfindung wird in dem CDMA-Übertragungssystem die Spreizcodierung zu  
15 einem anderen Zeitpunkt gestartet als die Signalübertragung, wobei der n Bit umfassende, N-fach iterierte Endzustand als Initialisierungszustand für die Erzeugung der zeitlich verschobenen Spreizfolge verwendet wird. Dies ermöglicht eine höhere Flexibilität beim zeitlichen Ablauf  
20 von Sende- und Empfangsvorgängen. Insbesondere kann der im Standard UMTS vorgesehene Compressed Mode mit geringem Aufwand implementiert werden.

Es ist von Vorteil, wenn durch eine gegebene Code-Nummer  
25 der Offset einer Spreizfolge festgelegt wird, wobei der n Bit umfassende, N-fach iterierte Endzustand als Initialisierungszustand für die Erzeugung der der Code-Nummer N zugeordneten Spreizfolge verwendet wird. Dadurch ist es möglich, eine große Zahl von Codes auf einfache Weise zu adressieren. Die Code-Nummer N, die zur Kennzeichnung eines Codes verwendet wird, dient gleichzeitig als maßgeblicher Parameter für die Codeerzeugung und kann direkt für die Codeerzeugung herangezogen werden. Zeitaufwendige Umrechnungen sind nicht erforderlich.  
30

Nachfolgend wird die Erfindung anhand von mehreren in der Zeichnung dargestellten Ausführungsbeispielen weiter beschrieben. Es zeigen:

- 5    Fig. 1    das Schaltbild eines Schieberegisters vom Typ SSRG (Simple Shift Register Generator);
- 10    Fig. 2    die erfindungsgemäße Darstellung der  $n \times n$  Matrix  $T^N$ , welche bei Multiplikation mit dem Anfangszustand direkt den N-fach iterierten Initialisierungszustand für die Erzeugung der verschobenen Pseudo-Noise-Folge liefert; und
- 15    Fig. 3    eine Tabelle, in der die Anzahl der benötigten Operationen in Abhängigkeit von dem gewünschten Offset N für das bisherige Verfahren und das erfindungsgemäße Verfahren miteinander verglichen werden.

20    In Fig. 1 ist die Struktur eines Schieberegisters vom Typ SSRG (Simple Shift Register Generator) gezeigt. Das Schieberegister umfasst  $n$  Registerzellen  $R_1, R_2, \dots, R_{n-1}, R_n$ , wobei der Registerinhalt einer Zelle jeweils die Werte 0 oder 1 annehmen kann. Über die gemeinsame Taktleitung 1 werden den Registerzellen Taktimpulse zugeführt. Mit jedem Taktimpuls wird der Inhalt einer Registerzelle in die

25    nächstfolgende Registerzelle übernommen. Hierzu ist der Ausgang einer Registerzelle jeweils mit dem Eingang der nächstfolgenden Registerzelle verbunden. Beispielsweise ist

30    der Ausgang der Registerzelle  $R_1$  über die Signalleitung 2 mit dem Eingang der Registerzelle  $R_2$  verbunden. Dadurch kann erreicht werden, dass die anfangs vorliegende Bitsequenz mit jedem Taktimpuls um eine Registerzelle bzw. um eine Bitposition nach rechts geschoben wird.

35

Das am Ausgang der Registerzelle  $R_n$  abgreifbare Signal 3 wird durch eine Anzahl von XOR-Gattern 4, 6, ..., 9, 11 mo-

difiziert, um das Signal 12 zu erhalten, das am Eingang der ersten Registerzelle  $R_1$  anliegt. Die Art und Weise, wie das am Ausgang von  $R_n$  abgreifbare Signal 3 modifiziert wird, um das Signal 12 zu erhalten, wird durch die Koeffizienten  $c_1, c_2, \dots, c_{n-2}, c_{n-1}$  festgelegt, welche jeweils den Wert 0 oder 1 annehmen können. Wenn  $c_i$  (mit  $i = 1, 2, \dots, n-1$ ) den Wert 0 hat, dann bedeutet dies, dass das am Ausgang der Registerzelle  $R_i$  abgreifbare Signal keinerlei Einfluss auf das Rückkopplungssignal hat. Wenn beispielsweise  $c_{n-1} = 0$  ist, dann wird das Signal 3 durch das am Ausgang der Registerzelle  $R_{n-1}$  abgreifbare Signal 13 nicht modifiziert. Das am ersten Eingang des XOR-Gatters 4 anliegende Signal 3 gelangt unverändert zum Ausgang des XOR-Gatters 4, so dass das Signal 5 dem Signal 3 entspricht. Wenn der Koeffizient  $c_{n-1} = 0$  ist, dann kann das XOR-Gatter 4 daher auch weggelassen und durch eine direkte Verbindung zwischen dem Signal 3 und dem Signal 5 ersetzt werden.

Wenn ein Koeffizient  $c_i$  (mit  $i = 1, 2, \dots, n-1$ ) dagegen gleich Eins ist, dann trägt das am Ausgang der Registerzelle  $R_i$  abgreifbare Signal zum rückgekoppelten Signal bei. Wenn beispielsweise  $c_2 = 1$  ist, dann wird das bisherige rückgekoppelte Signal 8 im XOR-Gatter 9 mit dem am Ausgang der Registerzelle  $R_2$  abgreifbaren Signal 14 XOR-verknüpft, so dass man das modifizierte rückgekoppelte Signal 10 erhält. Da eine XOR-Verknüpfung als Modulo-Zwei-Addition beschrieben werden kann, sind die XOR-Gatter 4, 6, ..., 9, 11 in Fig. 1 als Modulo-Zwei-Addierer eingezeichnet.

Die Rekursionsvorschrift für ein Schieberegister der in Fig. 1 gezeigten Art wird durch ein charakteristisches Polynom der Form

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

35

vorgegeben, wobei die Koeffizienten  $c_1, c_2, \dots, c_{n-1}$  den in Fig. 1 eingezeichneten Koeffizienten entsprechen und inso-

fern die Werte 0 oder 1 annehmen können. Im Bereich der Codierung bzw. Decodierung von Signalen werden als Polynome  $f(x)$  irreduzible Polynome verwendet. Irreduzible Polynome sind dadurch gekennzeichnet, dass sie sich nicht als Produkt von mindestens zwei Faktoren darstellen lassen, die ihrerseits auch Polynome mit einem Grad größer 0 über dem Körper  $GF(2)$  sind. Irreduzible Polynome lassen sich also nicht in Polynome von niedrigerem Grad faktorisieren.

- 10 Zum Zeitpunkt Null seien die Anfangswerte der Registerzellen  $R_1, R_2, \dots, R_n$  gegeben als  $x_1(0), x_2(0), \dots, x_n(0)$ . Die Werte der Register  $x_1(t+1), x_2(t+1), \dots, x_n(t+1)$  zum Zeitpunkt  $t+1$  lassen sich jeweils aus den Werten der Register  $x_1(t), x_2(t), \dots, x_n(t)$  zum Zeitpunkt  $t$  anhand folgender
- 15 Rekursionsvorschrift herleiten:

$$\begin{aligned} x_n(t+1) &= x_{n-1}(t), \\ x_{n-1}(t+1) &= x_{n-2}(t), \\ &\vdots \\ x_2(t+1) &= x_1(t), \\ x_1(t+1) &= c_1 \cdot x_1(t) + c_2 \cdot x_2(t) + \dots + c_{n-1} \cdot x_{n-1}(t) + x_n(t). \end{aligned}$$

- Bei der hier verwendeten Addition handelt es sich um eine
- 20 Modulo-Zwei-Addition, also um eine XOR-Operation. Wenn es sich bei  $f(x)$  um ein irreduzibles Polynom handelt, dann kann am Ausgang des Schieberegisters als Signal 3 eine sogenannte Pseudo-Noise-Folge

- 25  $x_n(0), x_n(1), x_n(2), x_n(3), \dots$

abgegriffen werden. Mit jedem Taktimpuls des Taktsignals 1 erscheint ein neuer Folgenwert am Ausgang des Schieberegisters.

30

Die Pseudo-Noise-Folgen, die sich mit der in Fig. 1 gezeigten Hardware erzeugen lassen, weisen geeignete Korrelationseigenschaften für die Signalcodierung auf. In CDMA-

Verfahren wie UMTS oder IS-95 werden daher derartige Pseudo-Noise-Folgen zur sender- und empfängerseitigen Erzeugung von Spreizfolgen verwendet. Die in Fig. 1 gezeigte Schieberegisterstruktur stellt daher die geeignete Hardware für

5 die Erzeugung von Spreizfolgen in Mobilstationen und Basisstationen dar, welche als Übertragungsstandard ein CDMA-Verfahren verwenden.

Der Registervektor

10

$$\begin{pmatrix} x_n(t) \\ x_{n-1}(t) \\ \vdots \\ x_2(t) \\ x_1(t) \end{pmatrix}$$

stellt den Inhalt der Registerzellen  $R_1, R_2, \dots, R_n$  zum Zeitpunkt  $t$  dar. Wenn man die  $n \times n$  Matrix  $T$  definiert als

15

$$T = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & & & 0 \\ \vdots & & & \ddots & & \vdots \\ & & & & 1 & 0 \\ 0 & & & & 0 & 1 \\ 1 & c_{n-1} & c_{n-2} & \dots & c_2 & c_1 \end{pmatrix},$$

dann lässt sich die Rekursionsvorschrift folgendermaßen formulieren:

20

$$\begin{pmatrix} x_n(t+1) \\ x_{n-1}(t+1) \\ \vdots \\ x_2(t+1) \\ x_1(t+1) \end{pmatrix} = T \cdot \begin{pmatrix} x_n(t) \\ x_{n-1}(t) \\ \vdots \\ x_2(t) \\ x_1(t) \end{pmatrix}.$$

Die  $n \times n$  Matrix  $T$  wird auch als charakteristische Matrix der Rekursion bezeichnet. Eine einmalige Iteration der Code-Folge kann also als Multiplikation der Matrix  $T$  mit dem Registervektor dargestellt werden. Entsprechend lässt sich  
 5 eine Verschiebung der Code-Folge um einen Offset  $N$  als Multiplikation des Registervektors mit der Matrix  $T^N$  darstellen:

$$\begin{pmatrix} x_n(t+N) \\ x_{n-1}(t+N) \\ \vdots \\ x_2(t+N) \\ x_1(t+N) \end{pmatrix} = T^N \cdot \begin{pmatrix} x_n(t) \\ x_{n-1}(t) \\ \vdots \\ x_2(t) \\ x_1(t) \end{pmatrix}.$$

10

Eine direkte Berechnung der  $N$ -ten Potenz der Matrix  $T$  wäre jedoch noch aufwendiger als die aus dem Stand der Technik bekannte Ausführung von  $N$  Vorab-Iterationen des Schieberegisters.

15

Im folgenden soll die Matrix  $T^N$  mit einem schnellen und wenig aufwendigen Verfahren bestimmt werden. Dabei soll von der  $n \times n$  Matrix  $T^*$  ausgegangen werden, welche die transponierte Matrix der Matrix  $T$  ist. Die Matrix  $T^*$  ist gegeben durch

$$T^* = \begin{pmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & c_{n-1} \\ 0 & 1 & 0 & & \vdots \\ 0 & \cdots & \ddots & 0 & c_2 \\ 0 & 0 & \cdots & 1 & c_1 \end{pmatrix}.$$

25 Die Erfindung beruht auf der Beobachtung, dass die Multiplikation mit der transponierten Matrix  $T^*$  einer Multiplikation mit der unabhängigen Variablen  $x$  im Restklassenring des Polynomrings modulo  $f^*$  entspricht. Das Polynom

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + c_1 \cdot x^{n-1} + x^n$$

wird dabei durch Spiegelung der Koeffizienten des Polynoms

5

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

erhalten. Man kann dies auch schreiben als

$$10 \quad f^*(x) = x^n \cdot f(x^{-1}).$$

Dass die Multiplikation mit  $T^*$  einer Multiplikation mit  $x$  modulo  $f^*$  entspricht, kann man sich folgendermaßen klar machen:

15

Jede Restklasse modulo  $f^*$  ist eine Linearkombination der „kanonischen Basis“  $[1], [x], \dots, [x^{n-1}]$  modulo  $f^*$ . Daher genügt es zu zeigen, dass  $T^*$  auf diese Basis so wirkt wie die Multiplikation mit  $x$  modulo  $f^*$ .

20

Die Äquivalenzklasse  $[1]$  modulo  $f^*$  ist gegeben durch den Vektor

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

25

Nach Multiplikation mit  $T^*$  erhält man den Vektor

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

dies entspricht der Äquivalenzklasse  $[x]$  modulo  $f^*$ . In derselben Weise gilt dies für alle Äquivalenzklassen  $[1], [x], \dots [x^{n-2}]$  modulo  $f^*$ . Die letzte Äquivalenzklasse  $[x^{n-1}]$  modulo  $f^*$  entspricht dem Vektor

5

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

dieser wird bei Multiplikation mit  $T^*$  abgebildet auf den Vektor

10

$$\begin{pmatrix} 1 \\ c_{n-1} \\ \vdots \\ c_2 \\ c_1 \end{pmatrix},$$

und dies entspricht der Äquivalenzklasse

$[1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + c_1 \cdot x^{n-1}] \bmod f^*$ . Aber diese Äquivalenzklasse ist genau die Äquivalenzklasse  $[x^n]$  modulo  $f^*$ , denn

$$\begin{aligned} [x^n] \bmod f^* &= \\ &= [x^n + f^*] \bmod f^* = \\ &= [x^n + 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + c_1 \cdot x^{n-1} + x^n] \bmod f^* = \\ &= [1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + c_1 \cdot x^{n-1}] \bmod f^* \end{aligned}$$

20 Hier bedeutet „+“ jeweils die Addition im entsprechenden Körper  $GF(2)$  mit zwei Elementen, d.h. das „+“ entspricht dem „XOR“.

Damit ist für jedes Basiselement die Multiplikation mit  $T^*$   
25 dasselbe wie die Multiplikation mit  $x$  modulo  $f^*$ , und daher



ist auch für jedes Polynom die Multiplikation mit  $T^*$  dasselbe wie die Multiplikation mit  $x$  modulo  $f^*$ .

5 Damit ist auch die Multiplikation mit  $(T^*)^N$  dasselbe wie die Multiplikation mit  $x^N$  modulo  $f^*$ .

Diese Eigenschaft kann ausgenutzt werden, um die Matrix  $(T^*)^N$  zu bestimmen. Die Matrix  $(T^*)^N$  beschreibt eine lineare Transformation, die das Polynom  $[x^{j-1}] \bmod f^*$  (mit  $j = 1, 2, \dots, n$ ) in das mit  $x^N$  modulo  $f^*$  multiplizierte Polynom  $[x^{N+j-1}] \bmod f^*$  überführt. Dabei wird das Polynom  $[x^{j-1}] \bmod f^*$ , genauer gesagt das die Restklasse  $[x^{j-1}] \bmod f^*$  repräsentierende Polynom mit Grad kleiner  $n$ , durch den  $j$ -ten Einheitsvektor dargestellt. Das Polynom  $[1] \bmod f^*$  wird also durch  
 15 den ersten Einheitsvektor

$$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

das Polynom  $[x] \bmod f^*$  durch den zweiten Einheitsvektor

20

$$\begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

dargestellt, etc. Die Multiplikation dieser Einheitsvektoren mit der Matrix  $(T^*)^N$  überführt den ersten Einheitsvektor in den Spaltenvektor  $[x^N] \bmod f^*$ , den zweiten Einheitsvektor in den Spaltenvektor  $[x^{N+1}] \bmod f^*$ , und allgemein den  $j$ -ten Einheitsvektor in den Spaltenvektor  $[x^{N+j-1}] \bmod f^*$ .  
 25 Daher besitzt die Matrix  $(T^*)^N$  die folgende Struktur:

$$(T^*)^N = \begin{pmatrix} [x^N] \bmod f^*, [x^{N+1}] \bmod f^*, \dots [x^{N+n-1}] \bmod f^* \end{pmatrix}.$$

Diese Notation bedeutet, dass die  $j$ -te Spalte der Matrix  $(T^*)^N$  durch die Koeffizienten von demjenigen Vertreter der Restklasse  $[x^{N+j-1}] \bmod f^*$  gebildet wird, der den kleinsten Grad aufweist. Wenn man an diese Matrix von rechts den  $j$ -ten Einheitsvektor heranmultipliziert, dann erhält man den gewünschten Spaltenvektor  $[x^{N+j-1}] \bmod f^*$ .

Bei der Matrix  $T$  dürfen die Operationen des Transponierens und des Potenzierens vertauscht werden. Es gilt also

$$(T^*)^N = (T^N)^*.$$

Daher ergibt sich für die zu bestimmende Matrix  $T^N$

$$T^N = (t_{j,k})_{j,k=1,2,\dots,n} = \begin{pmatrix} [x^N] \bmod f^* \\ [x^{N+1}] \bmod f^* \\ \vdots \\ [x^{N+n-1}] \bmod f^* \end{pmatrix}.$$

Die  $j$ -te Zeile der Matrix  $T^N$  wird durch die Koeffizienten von demjenigen Vertreter der Restklasse  $[x^{N+j-1}] \bmod f^*$  gebildet, der den kleinsten Grad aufweist. Diese Struktur der Matrix  $T^N$  ist in Fig. 2 dargestellt.

Damit ist die Berechnung der Matrix  $T^N$  abgeschlossen.

Die so ermittelte Matrix  $T^N$  kann nun in die Iterationsvorschrift

$$\begin{pmatrix} x_n(t+N) \\ x_{n-1}(t+N) \\ \vdots \\ x_2(t+N) \\ x_1(t+N) \end{pmatrix} = T^N \cdot \begin{pmatrix} x_n(t) \\ x_{n-1}(t) \\ \vdots \\ x_2(t) \\ x_1(t) \end{pmatrix}$$

eingesetzt werden. Für die Iterationsvorschrift zur Berechnung des N-fach iterierten Zustands ergibt sich damit:

5

$$\begin{pmatrix} x_n(t+N) \\ x_{n-1}(t+N) \\ \vdots \\ x_2(t+N) \\ x_1(t+N) \end{pmatrix} = \begin{pmatrix} [x^N] \bmod f^* \\ [x^{N+1}] \bmod f^* \\ \vdots \\ [x^{N+n-1}] \bmod f^* \end{pmatrix} \cdot \begin{pmatrix} x_n(t) \\ x_{n-1}(t) \\ \vdots \\ x_2(t) \\ x_1(t) \end{pmatrix}$$

Zur Berechnung der Matrixelemente  $(t_{j,k})_{k=1,2,\dots,n}$  der j-ten Zeile der Matrix  $T^N$  müssen die Koeffizienten von demjenigen Polynom bestimmt werden, das einerseits zur Restklasse  $[x^{N+j-1}] \bmod f^*$  gehört und andererseits einen Grad kleiner als n besitzt. Dies kann von einem sogenannten Square-and-Multiply Algorithmus geleistet werden. Derartige Algorithmen können ausgehend von dem Restklassenpolynom  $g = [x] \bmod f^*$ , das als Eingangsgröße des Algorithmus dient, das Restklassenpolynom  $[x^M] \bmod f^*$  bestimmen, wobei M eine beliebige natürliche Zahl ist.

15

Es sei  $M = M_r M_{r-1} M_{r-2} \dots M_1 M_0$  eine Binärdarstellung der natürlichen Zahl M, wobei das höchstwertige Bit  $M_r = 1$  ist. Dann schreibt sich der entsprechende Square-and-Multiply Algorithmus folgendermaßen:

20

1. Set  $y \leftarrow g$
- 25 2. For i from r-1 downto 0 do
  - 2.1 Set  $y \leftarrow y^2 \bmod f^*$
  - 2.2 If  $M_i = 1$  then set  $y \leftarrow g \cdot y \bmod f^*$

## 3. Output y

In Zeile 2.1 wird die Square-Operation, und in Zeile 2.2, falls  $M_i = 1$  ist, die Multiply-Operation durchgeführt. Der  
 5 Operator „•“ bezeichnet dabei die Multiplikation zweier Restklassen und liefert einen Vertreter der resultierenden Restklasse. Nach vollständiger Durchführung des Algorithmus erhält man als Output y den Vertreter der Restklasse  $[x^M] \bmod f^*$  mit dem niedrigsten Grad. Die Zahl der benötigten  
 10 Rechenschritte und damit auch die benötigte Rechenzeit hängt logarithmisch von M ab.

Entsprechend einer ersten Ausführungsform der Erfindung werden die Matrixelemente der Matrix  $T^N$  bestimmt, indem für  
 15 jede Zeile einmal der Square-and-Multiply Algorithmus ausgeführt wird. Zur Berechnung der Matrixelemente der j-ten Zeile, welche durch die Koeffizienten des Restklassenpolynoms  $[x^{N+j-1}] \bmod f^*$  gegeben ist, wird also der Square-and-Multiply Algorithmus für  $M = N+j-1$  aufgerufen. Durch n-  
 20 maliges Ausführen des Square-and-Multiply Algorithmus lassen sich so sämtliche Matrixelemente bestimmen.

Alternativ dazu werden gemäß einer zweiten Ausführungsform der Erfindung lediglich die Matrixelemente  $(t_{1,k})_{k=1,2,\dots,n}$  der  
 25 ersten Zeile der Matrix mittels des Square-and-Multiply Algorithmus bestimmt, während die Matrixelemente der Zeilen 2 bis n durch Iterieren der Matrixelemente der ersten Zeile gewonnen werden. Bei dieser Ausführungsform der Erfindung muss der Square-and-Multiply Algorithmus nur einmal auf-  
 30 rufen werden. Dadurch kann bei dieser Ausführungsform der Erfindung der Rechenaufwand weiter verringert werden.

Zunächst wird also der Square-and-Multiply Algorithmus für  $M = N$  aufgerufen, um die erste Zeile  $(t_{1,1}, t_{1,2}, \dots, t_{1,n-1}, t_{1,n})$   
 35 der Matrix  $T^N$  zu bestimmen. Diese Zeile besteht aus den Koeffizienten des Vertreters der Restklasse  $[x^N] \bmod f^*$ , es gilt also

$$[x^N] \bmod f^* = [t_{1,1} + t_{1,2} \cdot x + t_{1,3} \cdot x^2 + \dots + t_{1,n} \cdot x^{n-1}] \bmod f^*.$$

Ausgehend von dieser ersten Zeile der Matrix  $T^N$  sollen nun  
 5 die folgenden Zeilen der Matrix iterativ bestimmt werden.  
 Für die Bestimmung der Matrixelemente der nächstfolgenden,  
 j-ten Zeile aus der vorhergehenden, (j-1)-ten Zeile müssen  
 jeweils zwei Schritte durchgeführt werden. In einem ersten  
 Schritt werden die Matrixelemente der (j-1)-ten Zeile um  
 10 eine Position nach rechts verschoben, was einer Multiplika-  
 tion mit x entspricht. Es gilt also für  $j = 2, 3, \dots, n$ :

$$(t_{j,1}, t_{j,2}, t_{j,3}, \dots, t_{j,n-1}, t_{j,n}) := (0, t_{j-1,1}, t_{j-1,2}, \dots, t_{j-1,n-1})$$

15 Dabei wird das letzte Element der (j-1)-ten Zeile, das Mat-  
 rixelemente  $t_{j-1,n}$ , aus der Matrix herausgeschoben. Falls  
 das Matrixelement  $t_{j-1,n}$  jedoch gleich 1 ist, wird durch  
 dieses Matrixelement  $t_{j-1,n}$  eine Rückkopplung und somit eine  
 Modifikation der Matrixelemente der j-ten Zeile bewirkt. Im  
 20 zweiten Schritt muss deshalb zunächst abgefragt werden, ob  
 $t_{j-1,n} = 1$  ist. Falls  $t_{j-1,n} = 1$  ist, wird eine XOR-Addition  
 des gespiegelten Polynoms  $f^*(x)$  und der im ersten Schritt  
 erhaltenen Matrixelemente j-ten Zeile  $(t_{j,1}, t_{j,2}, \dots, t_{j,n-1}, t_{j,n})$   
 durchgeführt. Das gespiegelte Polynom

25 
$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

lässt sich auch schreiben als

30 
$$f^*(x) = f_1 + f_2 \cdot x + f_3 \cdot x^2 + \dots + f_{n+1} \cdot x^n$$

und kann daher durch den Bitvektor  $(f_1, f_2, \dots, f_{n-1}, f_n, f_{n+1})$   
 dargestellt werden. Für den Fall  $t_{j-1,n} = 1$  ist daher fol-  
 gende XOR-Addition durchzuführen:

35 
$$t_{j,k} := t_{j,k} \oplus f_k,$$

wobei  $k = 1, 2, \dots, n$  die verschiedenen Elemente der  $j$ -ten Zeile bezeichnet, und wobei der Operator „ $\oplus$ “ für die XOR-Addition steht.

- 5 Auf diese Weise können sämtliche Matrixelemente der Matrix

$$T^N = (t_{j,k})_{j,k=1,2,\dots,n}$$

bestimmt werden.

10

Die beiden Schritte des Rechtsverschiebens und der XOR-Addition von  $f^*$  entsprechen dabei genau den Operationen, die ein Schieberegister vom Typ MSRG (Modular Shift Register Generator) pro Taktimpuls durchführen würde. Die zur  
15 Bestimmung der Matrixelemente notwendigen Iterationen werden aber rein rechnerisch mittels eines Prozessors durchgeführt.

20

Eine der wichtigsten Anwendungen der Erfindung liegt in der Erzeugung von Spreizfolgen für Übertragungssysteme, welche entsprechend einem CDMA-Übertragungsverfahren arbeiten. Bei diesen Spreizfolgen handelt es sich um Pseudo-Noise-Folgen, welche entweder von einer Schieberegisteranordnung vom Typ SSRG, oder aber von einem digitalen Signalprozessor erzeugt  
25 werden.

30

Mit Hilfe der Erfindung ist es möglich, rechnerisch den Inhalt der Schieberegisteranordnung zu bestimmen, der sich nach Durchführung von  $N$  Iterationen ergeben würde. Dieser  
um  $N$  Bit verschobene Initialisierungszustand kann dann in die Registerzellen der Schieberegisteranordnung eingeschrieben werden. Ausgehend von diesem Initialisierungszustand erzeugt die Schieberegisteranordnung dann eine um  $N$  Bits verschobene Pseudo-Noise-Folge, welche als Spreizfolge  
35 eingesetzt werden kann.

Die Definitionen der für den UMTS-Mobilfunk zu verwendenden Codierungen finden sich in „3GPP: Spreading and modulation (FDD)“, 3<sup>rd</sup> Generation Partnership Project TS 25.213, Release 1999. Unter anderem werden hier die sogenannten

5 Scrambling-Codes definiert, mit denen die ausgesendeten Signale codiert werden. Diese Scrambling-Codes dienen unter anderem zur Unterscheidung der Signale, welche von verschiedenen Basisstationen zu einer Mobilstation gesendet werden (Downlink). Dabei werden im Downlink-Modus, also

10 beim Senden eines Signals von der Basisstation zur Mobilstation andere Codes verwendet als beim Senden eines Signals vom Mobilfunkbenutzer zur Basisstation (Uplink). Darüber hinaus werden die verschiedenen logischen Kanäle, etwa zur kontinuierlichen Daten-/Sprachübertragung, zur gebündelten Übertragung von Daten als Pakete und zum Abgleich

15 zwischen Sender und Empfänger, mit verschiedenen Scrambling-Codes codiert. Dabei kann jeweils aus einer Familie von Codes ausgewählt werden, wobei die Codes innerhalb einer Familie durch ihre Code-Nummern unterschieden

20 werden.

Im wesentlichen existieren in UMTS drei verschiedene Typen von Scrambling-Codes, welche jeweils aus einer Folge von komplexen Zahlen bestehen. Die sogenannten langen Codes bestehen aus 38400 Zahlen und weisen innerhalb eines Zeitrahmens von 10 ms keine Wiederholungen auf. Daneben gibt es die sogenannten kurzen Codes, welche sich alle 256 Zeichen wiederholen, sowie die sogenannten Preamble-Codes, welche aus 4096 bestehen. Die langen Scrambling-Codes weisen die

25 höchste Komplexität auf. Sie sind im UMTS-Standard mit Hilfe von Pseudo-Noise-Folgen definiert. Im Downlink-Modus, also beim Senden eines Signals von der Basisstation zur Mobilstation, werden zwei verschiedene Pseudo-Noise-Folgen verwendet, wobei die zugehörigen irreduziblen Polynome den

30 Grad 18 besitzen und gegeben sind durch  $f(x) = 1 + x^7 + x^{18}$  und  $f(x) = 1 + x^5 + x^7 + x^{10} + x^{18}$ .

35

Für den Fall, dass kein Offset vorgesehen ist, wird der Anfangszustand, also die anfängliche Registerbelegung der Schieberegisteranordnung, durch die technische Spezifikation des 3<sup>rd</sup> Generation Partnership Project explizit vorgegeben. Der Scrambling-Code mit Nummer N ergibt sich aus diesem Code durch Berücksichtigung eines zusätzlichen Offsets von N Bits.

Bei Verwendung eines Square-and-Multiply Verfahrens zur Berechnung von Restklassen in Polynomringen lässt sich das erfindungsgemäße Verfahren zur Bestimmung eines N-fach iterierten Zustands unter alleiniger Verwendung von Shift-Operationen implementieren. Das Verfahren des Stands der Technik, also die Abarbeitung von N Vorab-Iterationen, kann ebenfalls mit Hilfe von Shift-Operationen realisiert werden.

Fig. 3 zeigt eine Tabelle, in der für verschiedene Werte des Offsets N die Zahl der beim bisherigen Verfahren benötigten Operationen (mittlere Spalte) sowie die Zahl der beim erfindungsgemäßen Verfahren benötigten Operationen (rechte Spalte) angegeben ist. In technischer Realisierung ist die Anzahl der benötigten Operationen in etwa proportional zur benötigten Zeit. Man erkennt, dass das bisherige Verfahren nur für sehr kleine Werte des Offsets N schnell genug ist. Ein großer Vorteil des neuen Verfahrens liegt darin, dass die Anzahl der benötigten Operationen logarithmisch vom gewünschten Offset N abhängt. Dies führt zu einer signifikanten Reduzierung des rechnerischen und zeitlichen Aufwands. Darüber hinaus lässt sich der rechnerische und zeitliche Aufwand vorab wesentlich besser kalkulieren als bei dem Verfahren des Stands der Technik. Gerade für Mobilfunkanwendungen, welche stets in Echtzeit ablaufen müssen, ist dies ein gravierender Vorteil.



## Patentansprüche

1. Verfahren zur Bestimmung eines n Bit umfassenden, N-fach iterierten Endzustands einer Schieberegisteranordnung aus  
 5 einem gegebenen, n Bit umfassenden Anfangszustand der Schieberegisteranordnung, wobei die Iterationsvorschrift für die Schieberegisteranordnung durch das charakteristische Polynom

$$10 \quad f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

mit  $c_1, c_2, \dots, c_{n-1} \in \{0; 1\}$  gegeben ist,

g e k e n n z e i c h n e t d u r c h folgende Schritte:

a) Bestimmen des Polynoms

$$15 \quad f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

durch Spiegelung der Koeffizienten des Polynoms

$$20 \quad f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n;$$

b) für  $j = 1, \dots, n$ , Bestimmen von demjenigen Vertreter der Restklasse

$$25 \quad [x^{N+j-1}] \bmod f^*,$$

dessen Grad kleiner als n ist;

c) Multiplizieren der Bitsequenz des Anfangszustands mit  
 30 einer Matrix, deren j-te Zeile beziehungsweise j-te Spalte für  $j = 1, \dots, n$  durch die Koeffizienten des in Schritt b) bestimmten Vertreters der Restklasse

$$[x^{N+j-1}] \bmod f^*$$

35

gegeben ist.

2. Verfahren nach Anspruch 1,  
 d a d u r c h g e k e n n z e i c h n e t, dass  
 die Vertreter der Restklassen

$$5 \quad [x^N] \bmod f^*, [x^{N+1}] \bmod f^*, \dots [x^{N+n-1}] \bmod f^*$$

jeweils explizit mittels eines geeigneten Algorithmus, ins-  
 besondere mittels eines Square-and-Multiply Algorithmus,  
 berechnet werden.

10

3. Verfahren nach Anspruch 1,  
 d a d u r c h g e k e n n z e i c h n e t, dass  
 lediglich der Vertreter der Restklasse

$$15 \quad [x^N] \bmod f^*$$

explizit mittels eines geeigneten Algorithmus, insbesondere  
 mittels eines Square-and-Multiply Algorithmus, berechnet  
 wird, und dass die Vertreter der Restklassen

20

$$[x^{N+j-1}] \bmod f^*$$

mit  $j = 2, \dots, n$  durch  $(n-1)$  rechnerisch durchgeführte Itera-  
 tionen aus den Koeffizienten des Vertreters der Restklasse

25

$$[x^N] \bmod f^*$$

erhalten werden.

30

4. Verfahren nach Anspruch 3,  
 d a d u r c h g e k e n n z e i c h n e t, dass  
 die Vertreter der Restklassen

$$[x^{N+j-1}] \bmod f^*$$

35

mit  $j = 2, \dots, n$  durch  $(n-1)$  rechnerisch durchgeführten Iterationen einer Schieberegisteranordnung vom MSRG-Typ aus den Koeffizienten des Vertreters der Restklasse

$$5 \quad \left[ x^N \right] \bmod f^*$$

erhalten werden, wobei die Iterationsvorschrift für die Schieberegisteranordnung durch das charakteristische Polynom

10

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

gegeben ist.

15 5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass der  $n$  Bit umfassende,  $N$ -fach iterierte Endzustand als Initialisierungszustand für die Erzeugung einer um  $N$  Bit verschobenen Pseudo-Noise-Folge verwendet wird.

20

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass der  $n$  Bit umfassende,  $N$ -fach iterierte Endzustand als Initialisierungszustand in eine  $n$  Schieberegisterzellen  $(R_1, R_2, \dots, R_n)$  umfassende Schieberegisteranordnung eingeschrieben wird.

25

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass es sich bei der Schieberegisteranordnung um eine  $n$  Schieberegisterzellen  $(R_1, R_2, \dots, R_n)$  umfassende Schieberegisteranordnung vom Typ SSRG handelt, deren Struktur durch das charakteristische Polynom

30

$$35 \quad f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

gegeben ist.

8. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, dass  
das Verfahren in CDMA-Übertragungssystemen, insbesondere in  
5 CDMA-Übertragungssystemen mit den Übertragungsstandards  
UMTS oder IS-95 zur Erzeugung einer Spreizfolge mit einem  
Offset von N Bits verwendet wird.

9. Verfahren nach Anspruch 8,  
10 dadurch gekennzeichnet, dass  
das Verfahren zur Erzeugung der im UMTS-Standard definier-  
ten Scrambling-Codes eingesetzt wird.

10. Verfahren nach Anspruch 8 oder Anspruch 9,  
15 dadurch gekennzeichnet, dass  
die Spreizfolge zur senderseitigen Spreizcodierung der ge-  
sendeten Signale verwendet wird.

11. Verfahren nach Anspruch 8 oder Anspruch 9,  
20 dadurch gekennzeichnet, dass  
die Spreizfolge zur empfängerseitigen Decodierung der emp-  
fangenen Signale eingesetzt wird.

12. Verfahren nach einem der Ansprüche 8 bis 11,  
25 dadurch gekennzeichnet, dass  
in dem CDMA-Übertragungssystem die Spreizcodierung zu einem  
anderen Zeitpunkt gestartet wird als die Signalübertragung,  
wobei der n Bit umfassende, N-fach iterierte Endzustand als  
Initialisierungszustand für die Erzeugung der zeitlich ver-  
30 schobenen Spreizfolge verwendet wird.

13. Verfahren nach einem der Ansprüche 8 bis 11,  
dadurch gekennzeichnet, dass  
durch eine gegebene Code-Nummer der Offset einer Spreizfol-  
35 ge festgelegt wird, wobei der n Bit umfassende, N-fach ite-  
rierte Endzustand als Initialisierungszustand für die Er-

zeugung der der Code-Nummer N zugeordneten Spreizfolge verwendet wird.

14. Vorrichtung zur Bestimmung eines n Bit umfassenden, N-fach iterierten Endzustands einer Schieberegisteranordnung aus einem gegebenen, n Bit umfassenden Anfangszustand der Schieberegisteranordnung, wobei die Iterationsvorschrift für die Schieberegisteranordnung durch das charakteristische Polynom

10

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

mit  $c_1, c_2, \dots, c_{n-1} \in \{0; 1\}$  gegeben ist, mit

- Mitteln zur Bestimmung des Polynoms

15

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

durch Spiegelung der Koeffizienten des Polynoms

20

$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n;$$

- Mitteln zur Restklassenbestimmung, welche für  $j = 1, \dots, n$  jeweils denjenigen Vertreter der Restklasse

25

$$\left[ x^{N+j-1} \right] \bmod f^*$$

bestimmen, dessen Grad kleiner als n ist,

- Mitteln zur Multiplikation der Bitsequenz des Anfangszustands mit einer Matrix, deren j-te Zeile beziehungsweise deren j-te Spalte für  $j = 1, \dots, n$  durch die Koeffizienten des Vertreters der Restklasse

30

$$\left[ x^{N+j-1} \right] \bmod f^*$$

35

gebildet wird, dessen Grad kleiner als n ist.

15. Vorrichtung nach Anspruch 14,

d a d u r c h g e k e n n z e i c h n e t, dass  
die Mittel zur Restklassenbestimmung die Vertreter der  
Restklassen

$$5 \quad \left[ x^N \right] \bmod f^*, \left[ x^{N+1} \right] \bmod f^*, \dots \left[ x^{N+n-1} \right] \bmod f^*$$

jeweils explizit mittels eines geeigneten Algorithmus, ins-  
besondere mittels eines Square-and-Multiply Algorithmus,  
berechnen.

10

16. Vorrichtung nach Anspruch 14,  
d a d u r c h g e k e n n z e i c h n e t, dass  
die Mittel zur Restklassenbestimmung lediglich den Vertre-  
ter der Restklasse

15

$$\left[ x^N \right] \bmod f^*$$

explizit mittels eines geeigneten Algorithmus, insbesondere  
mittel eines Square-and-Multiply Algorithmus, berechnen,  
20 und dass die Mittel zur Restklassenbestimmung die Vertreter  
der Restklassen

$$\left[ x^{N+j-1} \right] \bmod f^*$$

25 mit  $j = 2, \dots, n$  durch  $(n-1)$  rechnerisch durchgeführte Ite-  
rationen aus den Koeffizienten des Vertreters der Restklas-  
se

$$\left[ x^N \right] \bmod f^*$$

30

erhalten.

17. Vorrichtung nach Anspruch 16,  
d a d u r c h g e k e n n z e i c h n e t, dass  
35 die Mittel zur Restklassenbestimmung die Vertreter der  
Restklassen

$$\left[ x^{N+j-1} \right] \bmod f^*$$

mit  $j = 2, \dots, n$  durch  $(n-1)$  rechnerisch durchgeführte Iterationen einer Schieberegisteranordnung von MSRG-Typ aus  
 5 den Koeffizienten des Vertreters der Restklasse

$$\left[ x^N \right] \bmod f^*$$

erhalten, wobei die Iterationsvorschrift für die Schieberegisteranordnung durch das charakteristische Polynom  
 10

$$f^*(x) = 1 + c_{n-1} \cdot x + c_{n-2} \cdot x^2 + \dots + x^n$$

gegeben ist.

15

18. Vorrichtung nach einem der Ansprüche 14 bis 17,  
 d a d u r c h g e k e n n z e i c h n e t, dass  
 die Vorrichtung zur Bestimmung eines  $n$  Bit umfassenden,  $N$ -  
 fach iterierten Endzustands den Endzustand als Initialisie-  
 20 rungszustand in eine  $n$  Schieberegisterzellen  $(R_1, R_2, \dots, R_n)$   
 umfassende Schieberegisteranordnung schreibt.

19. Vorrichtung nach Anspruch 18,

d a d u r c h g e k e n n z e i c h n e t, dass  
 25 es sich bei der Schieberegisteranordnung um eine  $n$  Schieberegisterzellen  $(R_1, R_2, \dots, R_n)$  umfassende Schieberegisteranordnung vom Typ SSRG handelt, deren Struktur durch das charakteristische Polynom

30 
$$f(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1} + x^n$$

gegeben ist.

20. Verwendung einer Vorrichtung nach einem der Ansprüche  
 35 14 bis 19 zur Erzeugung einer Spreizfolge mit einem Offset  
 mit  $N$  Bits in einem CDMA-Übertragungssystem, insbesondere

in einem CDMA-Übertragungssystem entsprechend einem der Übertragungsstandards UMTS oder IS-95.

21. Verwendung nach Anspruch 20,

- 5     d a d u r c h   g e k e n n z e i c h n e t, dass  
die Spreizfolge zur senderseitigen Spreizcodierung der zu  
sendenden Signale eingesetzt wird.

22. Verwendung nach Anspruch 20,

- 10    d a d u r c h   g e k e n n z e i c h n e t, dass  
die Spreizfolge zur empfängerseitigen Decodierung der emp-  
fangenen Signale eingesetzt wird.

23. Verwendung nach einem der Ansprüche 20 bis 22,

- 15    d a d u r c h   g e k e n n z e i c h n e t, dass  
in dem CDMA-Übertragungssystem die Spreizcodierung zu einem  
anderen Zeitpunkt gestartet wird als die Signalübertragung,  
wobei der n Bit umfassende, N-fach iterierte Endzustand als  
Initialisierungszustand für die Erzeugung der zeitlich ver-  
20    schobenen Spreizfolge eingesetzt wird.

24. Verwendung nach einem der Ansprüche 20 bis 22,

- 25    d a d u r c h   g e k e n n z e i c h n e t, dass  
durch eine gegebene Code-Nummer der Offset einer Spreizfol-  
ge festgelegt wird, wobei der n Bit umfassende, N-fach ite-  
rierte Endzustand als Initialisierungszustand für die Er-  
zeugung der der Code-Nummer N zugeordneten Spreizfolge ein-  
gesetzt wird.



## Zusammenfassung

Verfahren und Vorrichtung zur Bestimmung von Initialisierungszuständen bei Pseudo-Noise-Folgen

5

Das erfindungsgemäße Verfahren ermöglicht die rechnerische Bestimmung eines  $n$  Bit umfassenden,  $N$ -fach iterierten Zustands einer Schieberegisteranordnung aus einem gegebenen Anfangszustand. Dadurch können Pseudo-Noise-Folgen mit beliebigem Offset  $N$  erzeugt werden, ohne dass hierzu Vorab-Iterationen durchgeführt werden müssten. Zur Berechnung des  $N$ -fach iterierten Endzustands wird eine Matrix herangezogen, deren  $j$ -te Zeile für  $j = 1, \dots, n$  durch die Koeffizienten desjenigen Vertreters der Restklasse  $[x^{N+j-1}] \bmod f^*$  gegeben ist, dessen Grad kleiner als  $n$  ist.

10

15

(Fig. 1)

1/2

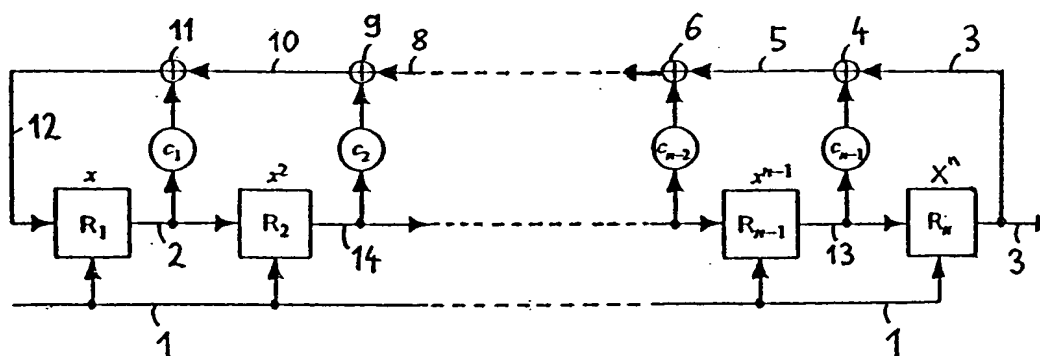


Fig. 1

$$T^N = \begin{pmatrix} (t_{1,k})_{k=1,2,\dots,n} \\ (t_{2,k})_{k=1,2,\dots,n} \\ \vdots \\ (t_{n,k})_{k=1,2,\dots,n} \end{pmatrix} = \begin{pmatrix} [x^N] \bmod f^* \\ [x^{N+1}] \bmod f^* \\ \vdots \\ [x^{N+n-1}] \bmod f^* \end{pmatrix}$$

Fig. 2

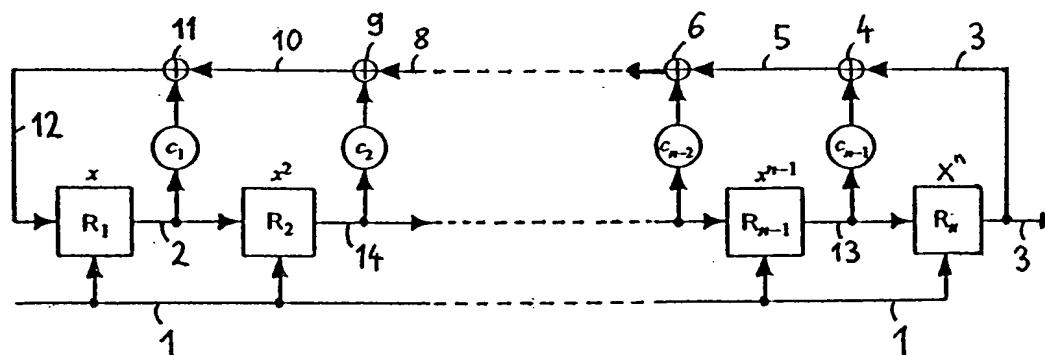


Fig. 1

$$T^N = \begin{pmatrix} (t_{1,k})_{k=1,2,\dots,n} \\ (t_{2,k})_{k=1,2,\dots,n} \\ \vdots \\ (t_{n,k})_{k=1,2,\dots,n} \end{pmatrix} = \begin{pmatrix} [x^N] \bmod f^* \\ [x^{N+1}] \bmod f^* \\ \vdots \\ [x^{N+n-1}] \bmod f^* \end{pmatrix}$$

Fig. 2

Gewünschter Offset	Bisheriges Verfahren	Oben beschriebenes Verfahren
100	100	660
1.000	1000	1.000
10.000	10.000	1.400
50.000	50.000	1.600
100.000	100.000	1.750
200.000	200.000	1.860

Fig. 3